Задача A. Заезд в ЛКШ

Имя входного файла:	`arrival.in`
Имя выходного файла:	`arrival.out`
Ограничение по времени:	1 секунда
Ограничение по памяти:	64 мегабайта

Знаете ли вы, как непросто организовать заезд в ЛКШ? Например, в 2012 году нужно было заказать автобусы для целых n ЛКШат, мечтающих добраться в "Берендеевы поляны" из Москвы. Один из директоров ЛКШ сообщил другому директору, что можно заказать некоторые из m автобусов. Он узнал вместимость каждого автобуса и сразу понял, какое минимальное количество автобусов ему нужно заказать, чтобы привезти в лагерь всех ЛКШат. А сможете ли вы так же быстро решить эту задачу?

Формат входных данных

В первой строке через пробел записаны целые числа n и m ($1 \leqslant n \leqslant 10^6$; $1 \leqslant m \leqslant 1000$). В следующей строке через пробел записаны m целых чисел в пределах от 1 до 1 000 — вместимости автобусов.

Формат выходных данных

В первой строке выведите число k — минимальное количество автобусов, которое придётся заказать директору. В следующей строке выведите через пробел k целых чисел — номера автобусов, которые нужно заказать. Автобусы пронумерованы от 1 до m в том порядке, в которых они перечислены во входных данных. Если возможных решений несколько, выведите любое. Если решения нет, в единственной строке выведите "`-1`".

Примеры

arrival.in	arrival.out
345 5 100 130 190 140 150	3 1 3 4
345 3 100 100 100	-1

Задача B. Такси

Имя входного файла:	`taxi.in`
Имя выходного файла:	`taxi.out`
Ограничение по времени:	1 секунда
Ограничение по памяти:	64 мегабайта

После затянувшегося совещания директор фирмы решил заказать такси, чтобы развезти сотрудников по домам. Он заказал N машин — ровно столько, сколько у него сотрудников. Но когда они подъехали, оказалось, что у каждого водителя такси свой тариф за 1 километр.

Каждый сотрудник сказал директору, сколько километров ему нужно проехать до дома. Разные сотрудники должны сесть в разные такси. Теперь директор хочет определить, какой из сотрудников на каком такси должен поехать домой, чтобы суммарные затраты на такси (а их несет фирма) были минимальны.

Формат входных данных

Сначала во входном файле записано натуральное число N ($1 \leqslant N \leqslant 1000$) — количество сотрудников компании (совпадающее с количеством вызванных машин такси). Далее записано N чисел, задающих расстояния в километрах от работы до домов сотрудников компании (первое число — для первого сотрудника, второе — для второго и т. д.). Все расстояния — положительные целые числа, не превышающие 1000. Далее записано еще N чисел — тарифы за проезд одного километра в такси (первое число — в первой машине такси, второе — во второй и т.д.). Тарифы выражаются положительными целыми числами, не превышающими 10000.

Формат выходных данных

В выходной файл выведите N чисел — оптимальное распределение сотрудников по такси. Первым выведите номер такси, в которое должен сесть первый сотрудник, вторым — номер такси, в которое должен сесть второй и т.д. Если есть несколько вариантов рассадки сотрудников, при которых затраты минимальны, выведите любой из них.

Примеры

taxi.in	taxi.out
3 10 20 30 50 20 30	1 3 2
5 10 20 1 30 30 3 3 3 2 3	1 2 3 5 4

Задача C. Тестирующая система

Имя входного файла:	`ejudge.in`
Имя выходного файла:	`ejudge.out`
Ограничение по времени:	1 секунда (python — 10 секунд)
Ограничение по памяти:	64 мегабайта

Юный программист Саша написал свою первую тестирующую систему. Он так обрадовался тому, что она скомпилировалась, что решил пригласить школьных друзей на свой собственный контест.

Но в конце тура выяснилось, что система не умеет сортировать команды в таблице результатов. Помогите Саше реализовать эту сортировку.

Команды упорядочиваются по правилам ACM:

- по количеству решённых задач в порядке убывания;
- при равенстве количества решённых задач — по штрафному времени в порядке возрастания;

- при прочих равных — по номеру команды в порядке возрастания.

Формат входных данных

Первая строка содержит натуральное число n ($1 \leqslant n \leqslant 100\,000$) — количество команд, участвующих в контесте. В i-й из следующих n строк записано количество решенных задач S ($0 \leqslant S \leqslant 100$) и штрафное время T ($0 \leqslant T \leqslant 100\,000$) команды с номером i.

Формат выходных данных

В выходной файл выведите n чисел — номера команд в отсортированном порядке.

Примеры

ejudge.in	ejudge.out
5 3 50 5 720 1 7 0 0 8 500	5 2 1 3 4

Задача D. Число

Имя входного файла:	`number.in`
Имя выходного файла:	`number.out`
Ограничение по времени:	1 секунда
Ограничение по памяти:	64 мегабайта

Вася написал на длинной полоске бумаги большое число и решил похвастаться своему старшему брату Пете этим достижением. Но только он вышел из комнаты, чтобы позвать брата, как его сестра Катя вбежала в комнату и разрезала полоску бумаги на несколько частей. В результате на каждой части оказалось одна или несколько идущих подряд цифр.

Теперь Вася не может вспомнить, какое именно число он написал. Только помнит, что оно было очень большое. Чтобы утешить младшего брата, Петя решил выяснить, какое максимальное число могло быть написано на полоске бумаги перед разрезанием. Помогите ему!

Формат входных данных

Входной файл содержит одну или более строк, каждая из которых содержит последовательность цифр. Количество строк во входном файле не превышает 100, каждая строка содержит от 1 до 100 цифр. Гарантируется, что хотя бы в одной строке первая цифра отлична от нуля.

Формат выходных данных

Выведите в выходной файл одну строку — максимальное число, которое могло быть написано на полоске перед разрезанием.

Примеры

number.in	number.out
2 20 004 66	66220004
3	3

Задача E. Минимум на стеке

Имя входного файла:	`stack.in`
Имя выходного файла:	`stack.out`
Ограничение по времени:	2 секунды (python — 10 секунд)
Ограничение по памяти:	256 мегабайт

Вам требуется реализовать структуру данных, выполняющую следующие операции:

1. Добавить элемент x в конец структуры.
2. Удалить последний элемент из структуры.
3. Выдать минимальный элемент в структуре.

Формат входных данных

В первой строке входного файла задано одно целое число n — количество операций ($1 \leqslant n \leqslant 10^6$). В следующих n строках заданы сами операции. В i–ой строке число t_i — тип операции (1, если операция добавления. 2, если операция удаления. 3, если операция минимума). Если задана операция добавления, то через пробел записано целое число x — элемент, который следует добавить в структуру ($-10^9 \leqslant x \leqslant 10^9$). Гарантируется, что перед каждой операцией удаления или нахождения минимума структура не пуста.

Формат выходных данных

Для каждой операции нахождения минимума выведите одно число — минимальный элемент в структуре. Ответы разделяйте переводом строки.

Примеры

stack.in	stack.out
8 1 2 1 3 1 -3 3 2 3 2 3	-3 2 2